

15

duplicate records containing all the name variations can be found, and perhaps pruned from the database.

Having thus described the invention, I claim:

1. A method of storing, in compressed form, a plurality of simultaneously occurring parallel data bodies, each comprised of sequentially ordered signals and with implied associations among the signals in the parallel data bodies, comprising the steps of:

creating a dictionary of unique signals for each of the simultaneously occurring parallel data bodies to be compressed and stored, the dictionary in each case consisting of a numerical index value associated with each unique field entry;

forming an n-ary tree having terminal and non-terminal nodes, including a single, highest non-terminal node representative of all lower derivative nodes, the terminal nodes of the tree corresponding to the dictionaries created for each of the simultaneously occurring parallel data bodies to be compressed and stored, each non-terminal node of the tree being represented by an associative memory assigning a numerical index value to each unique combination of the index values of the two nodes from which that non-terminal node is derived;

ordering the elements of the n-ary nodes by frequency and, thence, canonically within each associative memory;

canonically ordering one or more of the "n" sets of derivative index values within each associative memory;

storing a signal indicative of the counts of the number of times a signal or group of signals occurred in the data body;

reducing the canonically ordered set of index values to binary form, alternating from one binary value to the other as a change occurs in the ordering; and

storing data representative of the associative memories having been ordered and reduced.

2. The method of claim 1, the simultaneously occurring parallel data bodies being representative of a database characterized in having tables of records comprised of elements called fields.

3. The method of claim 1, further including the step of pre-compressing one or more of the simultaneously occurring parallel data bodies using an N-gram data storage system.

4. The method of claim 1, further including the step of taking a random sample of the simultaneously occurring parallel data bodies to be compressed in order to determine an ordering and composition of the nodes in the n-ary tree.

16

5. The method of claim 1, further including the step of hashing the input signals into linked lists comprising the nodes of the n-ary tree.

6. The method of claim 1, further including the step of creating indices indicative of the locations where there are stored signals relating to the current storage at the node immediately above a given node.

7. The method of claim 1, further including the step of creating indices indicative of the locations where there are stored signals relating to the current storage at the node immediately below a given node.

8. The method of claim 1, further including the step of creating indices indicative of another location in a given node storing the same signals from a derivative node.

9. The method of claim 1, wherein additional counts are kept which relate to the number of occurrences of each of the signals described at a given node.

10. The method of claim 1, where the elements of the nodes are searched for matches to a previously identified signal set and a signal indicative of at least the relative success of the match through a specified number of levels.

11. The method of claim 1, wherein the step of canonically ordering one of the two sets of derivative index values within each associative memory includes canonically ordering the set representative of the larger of the two memories from which that non-terminal node is derived.

12. The method of claim 1, wherein the step of forming an n-ary tree having terminal and non-terminal nodes includes the step of combining signals which exhibit a high degree of correlation therebetween.

13. The method of claim 1, wherein the step of forming an n-ary tree includes the step of locating fields with large dictionaries high in the tree.

14. The method of claim 1, wherein the step of forming an n-ary tree includes the step of arranging the tree such one of the two derivative memories for a particular non-terminal node is much larger than the other.

15. The method of claim 1, wherein the step of canonically ordering one or more of the sets of derivative index values within each associative memory includes canonically ordering the set representative of the largest of the derivative memories.

16. The method of claim 1, wherein the step of storing data representative of the associative memories having been ordered and reduced includes the step of storing all data except that associated with the highest non-terminal node or nodes in a type of memory have a faster access time as compared to the type of memory used to store the data associated with the highest non-terminal node or nodes.

* * * * *